Kindelan, M., Moscoso, M., & González-Rodríguez, P. (2016). Radial basis function interpolation in the limit of increasingly flat basis functions. *Journal of Computational Physics*, 307, 225-242.

DOI: 10.1016/j.jcp.2015.12.015

Radial basis function interpolation in the limit of increasingly flat basis functions

Manuel Kindelan, Miguel Moscoso, and
Pedro González-Rodríguez *

Abstract

We propose a new approach to study Radial Basis Function (RBF) interpolation in the limit of increasingly flat functions. The new approach is based on the semi-analytical computation of the Laurent series of the inverse of the RBF interpolation matrix described in a previous paper [3]. Once the Laurent series is obtained, it can be used to compute the limiting polynomial interpolant, the optimal shape parameter of the RBFs used for interpolation, and the weights of RBF finite difference formulas, among other things.

1 Introduction

This paper is concerned with the behavior of Radial Basis Function (RBF) interpolation in the limit of increasingly flat functions. In the past, there has been a considerable interest in analyzing this limit [6,9,11,12,15,18,20,21,27] since it leads to accurate interpolants which are effective both for interpolation problems and for solving partial differential equations.

Many RBFs commonly used in interpolation contain a *shape parameter* $\epsilon > 0$ which controls their flatness. As $\epsilon \to 0$, the RBF becomes increasingly flat. In this limit, the interpolation system becomes highly ill-conditioned, but the limit RBF interpolant at any point is well behaved so it converges to a finite number (except in some particular cases). Indeed, Driscoll and Fornberg [6] proved that 1D RBF interpolants converge to Lagrange interpolating polynomials, subject to some easily stated conditions on the RBF. They also observed numerically that in 2D the situation is more complicated, as the limit may

* Corresponding author. Address: Universidad Carlos III de Madrid, Avenida de la Universidad 30, 28911 Leganés, Spain. Fax: +34 91 624 91 29
Email addresses: moscoso@math.uc3m.es, kinde@ing.uc3m.es, pgonzale@ing.uc3m.es (Pedro González-Rodríguez).

not exist and, if it exists, it is a multivariate polynomial that might depend on the node layout and on the used RBF. Existence of the limit polynomial was proved in [19,28]. Conditions on the used RBFs, so that multivariate interpolation converges, have been recently derived in [20,21].

In this work, we analyze the limit of flat RBFs using the framework proposed in [3]. The main ingredient used in our analysis is the Laurent series of the inverse of the interpolation matrix, which we compute using a semi-analytical procedure [3]. The relevant parameter in the Laurent series is $\delta = (\epsilon h)^2$, which is the square of the product of the shape parameter ϵ and and a characteristic inter-nodal distance h. If we denote by $r_{i,j}$ the distamce between nodes i and j, then the dimensionless distances $r_{i,j}/h$ are of order unity (for details, see, for example, [32]). Multiplying the Laurent series of the inverse of the interpolation matrix by the data at the nodes we obtain a Laurent series for the interpolation coefficients avoiding the ill-conditioning associated to straightforward numerical approaches in the flat RBF limit.

In reference [2] we use the Laurent series of the inverse to compute the weights of RBF-FD formulas. In this paper we use the Laurent series of the inverse to approximate the RBF interpolant by a series of interpolation polynomials. This approach has several advantages for different issues related to RBF interpolation. We focus our attention in the following three important issues described below where we also note which contributions are novel:

- Derivation of the interpolating polynomial, which is the limit of RBF interpolation when $\delta \to 0$. These polynomials have been derived in some specific cases using symbolic language [6,18]. We not only obtain the leading order polynomial but a series of polynomials in powers of δ. Furthermore, we derive them for any node layout.
- Computation of the optimal value of the shape parameter in RBF interpolation. We propose a new method that makes use of the first two terms of the series of polynomials to obtain the value of δ that minimizes the interpolation error. We consider this new method as one of the main contributions of the paper.
- Derivation of RBF finite difference (RBF-FD) formulas. We use the series of polynomials in powers of δ to obtain the weights of RBF-FD formulas. In this way, we obtain formulas for each weight as a series in powers of δ. We also use these weights to derive exact formulas for the local truncation error.

The results presented in the paper show the usefulness of the Laurent series of the inverse to analyze RBF interpolation in the flat limit. It should be emphasized, though, that for large size problems the bottleneck is the computation of the Laurent series. The semi-analytical procedure that we use [3] is very accurate and efficient compared to its symbolic computation. However, its computational cost grows exponentially with the order of the singularity of the Laurent series. Thus, we can only compute stencils with singularities whose order is not greater than seven. Since the order of the singularity grows with the number of nodes [2] this means that the method is only applicable to a relatively small number of nodes. In fact, it is only possible to compute the Laurent series of the inverse for 36 nonequispaced nodes or 24 equispaced nodes in 2D, and for 84 nonequispaced nodes or 31 equispaced nodes in 3D. To apply it to large number of nodes would require a much faster procedure to compute the Laurent series.

The paper is organized as follows: Section 2 describes the formulation based on the Laurent series of the inverse and how to compute a Laurent series of polynomials that approximates the RBF interpolant in the limit $\delta \to 0$. Section 3 describes several significant results obtained with the proposed procedure. It is structured into three subsections focused on three main applications: limiting polynomial interpolant, computation of the optimal shape parameter and derivation of RBF-FD formulas. Finally, Section 4 contains the main conclusions of the paper.

2 Formulation

RBF interpolation is a very efficient technique for the approximation of scattered data. The data is approximated in the functional space spanned by a set of translated RBFs $\phi(\| \mathbf{x} - \mathbf{x}_k \|)$, where $\phi(\hat{r})$ is a function that only depends on the distance $\hat{r}_k = \| \mathbf{x} - \mathbf{x}_k \|$ to a node $\mathbf{x}_k$. RBFs often contain a free parameter which greatly influences the accuracy of the RBF approximation. For instance, in the case of multiquadrics ($\phi(\hat{r};\epsilon) = \sqrt{1+\epsilon^2\,\hat{r}^2}$) or gaussians ($\phi(\hat{r};\epsilon) = e^{-\epsilon^2\,\hat{r}^2}$) the free parameter ϵ, known as *shape* parameter, determines the flatness of the radial basis function; as $\epsilon \to 0$ these functions become increasingly flat near the origin. It is convenient to use dimensionless distances by using a characteristic internodal distance h as the spatial unit. Thus, $\epsilon^2 \hat{r}_k^2 = \epsilon^2\, h^2\, \frac{\|\mathbf{x}-\mathbf{x}_k\|^2}{h^2} = \delta\, r_k^2$, where $\delta = (\epsilon h)^2$, is the square of the product of the shape parameter ϵ times the inter-nodal distance h, and r_k is the dimensionless distance $\| \mathbf{x} - \mathbf{x}_k \| /h$. With this notation the RBF $\phi(\hat{r};\epsilon)$ is rewritten as $\phi(r;\delta)$.

If the data are given at n nodes $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n$ in d dimensions, the RBF interpolant is given by

$$s(\mathbf{x};\delta) \;=\; \sum_{k=1}^{n} \alpha_k(\delta)\, \phi\left(\frac{\| \mathbf{x} - \mathbf{x}_k \|}{h};\delta\right) \;=\; \sum_{k=1}^{n} \alpha_k(\delta)\, \phi(r_k;\delta)\,, \qquad (1)$$

where r_k is the nondimensional distance to node $\mathbf{x}_k$. For given data values

$f_i = f(\mathbf{x}_i)$, the interpolation coefficients α_k are obtained by solving the linear system

$$A(\delta)\,\boldsymbol{\alpha}(\delta) \;=\; \mathbf{f}\,, \tag{2}$$

where the entries of the $n\times n$ interpolation matrix are $A_{i,j} = \phi\left(\frac{\|\,\mathbf{x}_i - \mathbf{x}_k\,\|}{h};\delta\right)$, and $\boldsymbol{\alpha}(\delta) = [\,\alpha_1(\delta)\;\alpha_2(\delta)\;\ldots\;\alpha_n(\delta)\,]^T$ and $\mathbf{f} = [\,f_1\;f_2\;\ldots\;f_n\,]^T$ are n-dimensional column vectors. Equation (2) implies that $s(\mathbf{x};\delta)$ computed in (1) interpolates $f(\mathbf{x})$ at nodes $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n$. For many choices of RBFs (including multiquadrics) the system is nonsingular for any arbitrary set of nodes. In fact, Micchelli [23] proved that a sufficient condition to guarantee the non-singularity is that the interpolation matrix is strictly positive definite. Furthermore, it is well known, that large values of δ lead to well-conditioned linear systems, but the resulting approximation is inaccurate. On the other hand, small values of δ lead to accurate results but the condition number of (2) grows rapidly and, hence, the interpolation coefficients α_k diverge in the limit $\delta \to 0$. However, it has been shown [6] that although the interpolation coefficients diverge, the RBF interpolant itself (1) converges to a finite limit (except in some particular cases). Thus, computing $s(\mathbf{x},\delta)$ from $\mathbf{f}(\mathbf{x})$ is a well-conditioned process, but the intermediate step of computing $\boldsymbol{\alpha}$ is ill-conditioned.

In this paper, we compute the interpolation coefficients by means of the Laurent series of the inverse of the interpolation matrix A, which we derive using the semi-analytical procedure described in [3] for infinitely smooth RBFs. In this way, we avoid the difficulties associated to the numerical solution of the ill-conditioned system (2). The algorithm takes as input the matrices A_k that determine the power series expansion of the interpolation matrix $A(\delta)$. That is, the $n \times n$ matrices A_k in

$$A(\delta) \;= A_0 + \sum_{k=1}^{\infty} \delta^k\, A_k\,. \tag{3}$$

The output of the algorithm are the matrices H_k which determine the Laurent series expansion of the inverse of the interpolation matrix. That is, the $n \times n$ matrices H_k in

$$A^{-1}(\delta) \;= \sum_{k=-s}^{\infty} \delta^k\, H_k\,, \tag{4}$$

where $H_{-s} \neq 0$, and s is the order of the singularity.

It is apparent that the Laurent series for the inverse (4) can be used to derive the Laurent series for the interpolation coefficients, namely

$$\boldsymbol{\alpha}(\delta) \;=\; A^{-1}(\delta)\,\mathbf{f} \;=\; \sum_{k=-s}^{\infty} \delta^k\, H_k\,\mathbf{f} \;=\; \sum_{k=-s}^{\infty} \delta^k \boldsymbol{\alpha}_k\,, \tag{5}$$

where $\boldsymbol{\alpha}_k = H_k\,\mathbf{f}$ are n-dimensional column vectors. According to (1), the value of the RBF interpolant at position $\mathbf{x}$ is obtained by multiplying the weights (5) by the corresponding RBFs evaluated at position $\mathbf{x}$. Furthermore, the Taylor series expansion of the Radial basis function $\phi(r_k;\delta)$ is

$$\phi(r_k;\delta) \;=\; a_0 \;+\; a_1\,\delta\,r_k^2 \;+\; a_2\,\delta^2\,r_k^4 \;+\; \ldots \;+ a_j\,\delta^j\,r_k^{2\,j} \;+ \ldots\,. \tag{6}$$

Table 1 shows the expansion coefficients for some popular infinitely smooth RBFs. Hence, using the expansion (6) in (1), we can easily obtain the RBF

Table 1
RBFs expansion coefficients

RBF	a_0	$a_j\,,j=1,\ldots$
Multiquadric (MQ)	1	$\frac{(-1)^{j+1}}{2j}\prod_{i=1}^{j-1}\frac{2i-1}{2i}$
Inverse Multiquadric (IM)	1	$(-1)^j\prod_{i=1}^{j}\frac{2i-1}{2i}$
Inverse Quadratic (IQ)	-1	$(-1)^j$

interpolation in the limit $\delta \to 0$ as a Laurent series of polynomials in powers of δ. Namely,

$$p(\mathbf{x};\delta) \;\equiv \lim_{\delta\to 0} s(\mathbf{x};\delta) \;=\; \boldsymbol{\alpha}(\delta)^T \begin{bmatrix} a_0 \;+\; a_1\,\delta\,r_1^2 \;+\; a_2\,\delta^2\,r_1^4 \;+\; \ldots \\ a_0 \;+\; a_1\,\delta\,r_2^2 \;+\; a_2\,\delta^2\,r_2^4 \;+\; \ldots \\ \ldots \quad \ldots \quad \ldots \\ a_0 \;+\; a_1\,\delta\,r_n^2 \;+\; a_2\,\delta^2\,r_n^4 \;+\; \ldots \end{bmatrix} =$$

$$= \frac{1}{\delta^s}\,p_{-s}(\mathbf{x}) \;+\; \frac{1}{\delta^{s-1}}\,p_{1-s}(\mathbf{x}) \;+\; \ldots \;+\; p_0(\mathbf{x}) \;+\; \delta\,p_1(\mathbf{x}) \;+\; \ldots\,. \tag{7}$$

For instance, the two first terms in the expansion are given by

$$p_{-s}(\mathbf{x}) = \boldsymbol{\alpha}_{-s}^T \begin{bmatrix} a_0 \\ a_0 \\ \cdots \\ a_0 \end{bmatrix}, \quad p_{1-s}(\mathbf{x}) = \boldsymbol{\alpha}_{1-s}^T \begin{bmatrix} a_0 \\ a_0 \\ \cdots \\ a_0 \end{bmatrix} + \boldsymbol{\alpha}_{-s}^T \begin{bmatrix} a_1\, r_1^2 \\ a_1\, r_2^2 \\ \cdots \\ a_1\, r_n^2 \end{bmatrix}. \tag{8}$$

In cases where the limiting RBF interpolant is convergent, all the singular terms in the expansion are identically zero and the first non zero term is $p_0(\mathbf{x})$ which is given by

$$p_0(\mathbf{x}) = \boldsymbol{\alpha}_0^T \begin{bmatrix} a_0 \\ a_0 \\ \cdots \\ a_0 \end{bmatrix} + \boldsymbol{\alpha}_{-1}^T \begin{bmatrix} a_1\, r_1^2 \\ a_1\, r_2^2 \\ \cdots \\ a_1\, r_n^2 \end{bmatrix} + \ldots + \boldsymbol{\alpha}_{-s}^T \begin{bmatrix} a_s\, r_1^{2s} \\ a_s\, r_2^{2s} \\ \cdots \\ a_s\, r_n^{2s} \end{bmatrix}. \tag{9}$$

3 Results

We start this Section by describing the proposed procedure in detail using a simple interpolation problem. Then, we include three subsections to discuss the three relevant applications of the method mentioned in the Introduction.

Let us consider RBF interpolation in 1D with three equispaced centers $[-1, 0, 1]$ using multiquadrics as RBFs [2]. For this simple example, the 1D RBF interpolation matrix is given by the 3×3 symmetric matrix

$$A(\delta) = \begin{bmatrix} 1 & \sqrt{1+\delta} & \sqrt{1+4\delta} \\ \sqrt{1+\delta} & 1 & \sqrt{1+\delta} \\ \sqrt{1+4\delta} & \sqrt{1+\delta} & 1 \end{bmatrix}. \tag{10}$$

It is apparent that $A(\delta)$ becomes singular in the limit $\delta \to 0$. Using the procedure described in [3], we derive the Laurent series of the inverse of the interpolation matrix which is given by

$$A^{-1}(\delta) = \frac{1}{\delta^2}\begin{bmatrix} -1/4 & 1/2 & -1/4 \\ 1/2 & -1 & 1/2 \\ -1/4 & 1/2 & -1/4 \end{bmatrix} + \frac{1}{\delta}\begin{bmatrix} -3/4 & 5/4 & -1/4 \\ 5/4 & -3 & 5/4 \\ -1/4 & 5/4 & -3/4 \end{bmatrix} + \begin{bmatrix} 0 & -1/16 & 1/2 \\ -1/16 & 0 & -1/16 \\ 1/2 & -1/16 & 0 \end{bmatrix} +$$

$$\delta\begin{bmatrix} -1/4 & 21/32 & -3/4 \\ 21/32 & -1 & 21/32 \\ -3/4 & 21/32 & -1/4 \end{bmatrix} + \delta^2\begin{bmatrix} 3/4 & -485/256 & 7/4 \\ -485/256 & 3 & -485/256 \\ 7/4 & -485/256 & 3/4 \end{bmatrix} + \dots \quad (11)$$

Notice that the inverse of the Laurent series diverges as δ^{-2}. Hence, the singularity is of order $s = 2$.

The Laurent series for the weights is computed using (11) in (5). Consider, for instance, the case in which the values of the function at the nodes are 0, 1, and 0. The Laurent series for the weights is

$$\boldsymbol{\alpha}(\delta) = \sum_{k=-2}^{\infty} \delta^k \boldsymbol{\alpha}_k = \frac{1}{\delta^2}\begin{bmatrix} 1/2 \\ -1 \\ 1/2 \end{bmatrix} + \frac{1}{\delta}\begin{bmatrix} 5/4 \\ -3 \\ 5/4 \end{bmatrix} + \begin{bmatrix} -1/16 \\ 0 \\ -1/16 \end{bmatrix} + \dots . \quad (12)$$

Hence, $\boldsymbol{\alpha}_{-2} = [1/2, -1, 1/2]^T$, $\boldsymbol{\alpha}_{-1} = [5/4, -3, 5/4]^T$, $\boldsymbol{\alpha}_0 = [-1/6, 0, -1/6]^T$, and so on. The value of the RBF interpolant at position x is obtained by multiplying the weights (12) by the corresponding multiquadrics basis functions, i.e.,

$$s(x;\delta) = \boldsymbol{\alpha}(\delta)^T \begin{bmatrix} \sqrt{1+\delta(x+1)^2} \\ \sqrt{1+\delta x^2} \\ \sqrt{1+\delta(x-1)^2} \end{bmatrix} . \quad (13)$$

In the limit $\delta \to 0$, we can expand (13) in powers of $\delta(x - x_k)^2$ using (6) with the expansion coefficients given in Table 1. This results in

$$p(x;\delta) \equiv \lim_{\delta\to 0} s(x;\delta) = \boldsymbol{\alpha}(\delta)^T \begin{bmatrix} 1 + \delta\,\frac{(x+1)^2}{2} - \delta^2\,\frac{(x+1)^4}{8} + \dots \\ 1 + \delta\,\frac{x^2}{2} - \delta^2\,\frac{x^4}{8} + \dots \\ 1 + \delta\,\frac{(x-1)^2}{2} - \delta^2\,\frac{(x-1)^4}{8} + \dots \end{bmatrix} . (14)$$

Thus,

$$p(x;\delta) = \frac{1}{\delta^2}\left\{\boldsymbol{\alpha}_{-2}^T\begin{bmatrix}1\\1\\1\end{bmatrix}\right\} + \frac{1}{\delta}\left\{\boldsymbol{\alpha}_{-1}^T\begin{bmatrix}1\\1\\1\end{bmatrix} + \boldsymbol{\alpha}_{-2}^T\begin{bmatrix}\frac{(x+1)^2}{2}\\ \frac{x^2}{2}\\ \frac{(x-1)^2}{2}\end{bmatrix}\right\} +$$

$$+\left\{\boldsymbol{\alpha}_0^T\begin{bmatrix}1\\1\\1\end{bmatrix} + \boldsymbol{\alpha}_{-1}^T\begin{bmatrix}\frac{(x+1)^2}{2}\\ \frac{x^2}{2}\\ \frac{(x-1)^2}{2}\end{bmatrix} + \boldsymbol{\alpha}_{-2}^T\begin{bmatrix}-\frac{(x+1)^4}{8}\\ -\frac{x^4}{8}\\ -\frac{(x-1)^4}{8}\end{bmatrix}\right\} + \dots . \qquad (15)$$

It is straightforward to check that, as expected, the terms of order δ^{-2} and δ^{-1} vanish and, therefore, the leading term of the limit polynomial is given by

$$p(x;\delta) \approx \boldsymbol{\alpha}_0^T\begin{bmatrix}1\\1\\1\end{bmatrix} + \boldsymbol{\alpha}_{-1}^T\begin{bmatrix}\frac{(x+1)^2}{2}\\ \frac{x^2}{2}\\ \frac{(x-1)^2}{2}\end{bmatrix} - \boldsymbol{\alpha}_{-2}^T\begin{bmatrix}\frac{(x+1)^4}{8}\\ \frac{x^4}{8}\\ \frac{(x-1)^4}{8}\end{bmatrix} = 1 - x^2 . \quad (16)$$

Thus, 1D RBF interpolation in the limit of increasingly flat basis functions coincides with Lagrange polynomial interpolation.

The same procedure can be used to determine the limit polynomial interpolant when the function at the nodes are arbitrary, so $[f_{-1}, f_0, f_1]^T$. In that case, the leading term is given by

$$p_0(x) = f_{-1}\frac{x(x-1)}{2} + f_0\,(1+x)(1-x) + f_1\frac{x(x+1)}{2}\,, \qquad (17)$$

which coincides with the Lagrange interpolating polynomial. We can also compute higher order terms in (7) and obtain

$$p_1(x) = \begin{bmatrix} -\frac{1}{2}\left(x^4 - \frac{1}{2}x^3 - x^2 + \frac{1}{2}x\right) \\ x^4 - x^2 \\ -\frac{1}{2}\left(x^4 + \frac{1}{2}x^3 - x^2 - \frac{1}{2}x\right) \end{bmatrix}^T \begin{bmatrix} f_{-1} \\ f_0 \\ f_1 \end{bmatrix}, \qquad (18)$$

$$p_2(x) = \begin{bmatrix} \frac{9}{16}x^6 - \frac{3}{16}x^5 + \frac{3}{8}x^4 - \frac{3}{8}x^3 - \frac{15}{16}x^2 + \frac{9}{16}x \\ -\frac{9}{8}x^6 - \frac{3}{8}x^4 + \frac{3}{2}x^2 \\ \frac{9}{16}x^6 + \frac{3}{16}x^5 + \frac{3}{8}x^4 + \frac{3}{8}x^3 - \frac{15}{16}x^2 - \frac{9}{16}x \end{bmatrix}^T \begin{bmatrix} f_{-1} \\ f_0 \\ f_1 \end{bmatrix}. \qquad (19)$$

Notice that p_1, p_2, ... are all zero at the interpolation nodes.

Let us consider now RBF interpolation in 2D using an equispaced five node stencil $[(0,0),(1,0),(0,1),(-1,0),(0,-1)]$. In this case, the RBF interpolation matrix is the 5×5 matrix

$$A(\delta) = \begin{bmatrix} 1 & \sqrt{\delta+1} & \sqrt{\delta+1} & \sqrt{\delta+1} & \sqrt{\delta+1} \\ \sqrt{\delta+1} & 1 & \sqrt{2\delta+1} & \sqrt{4\delta+1} & \sqrt{2\delta+1} \\ \sqrt{\delta+1} & \sqrt{2\delta+1} & 1 & \sqrt{2\delta+1} & \sqrt{4\delta+1} \\ \sqrt{\delta+1} & \sqrt{4\delta+1} & \sqrt{2\delta+1} & 1 & \sqrt{2\delta+1} \\ \sqrt{\delta+1} & \sqrt{2\delta+1} & \sqrt{4\delta+1} & \sqrt{2\delta+1} & 1 \end{bmatrix}. \tag{20}$$

The first three terms of the Laurent series of the inverse are

$$A^{-1}(\delta) \approx \frac{1}{\delta^2}\begin{bmatrix} -\frac{4}{3} & \frac{1}{3} & \frac{1}{3} & \frac{1}{3} & \frac{1}{3} \\ \frac{1}{3} & -\frac{1}{3} & \frac{1}{6} & -\frac{1}{3} & \frac{1}{6} \\ \frac{1}{3} & \frac{1}{6} & -\frac{1}{3} & \frac{1}{6} & -\frac{1}{3} \\ \frac{1}{3} & -\frac{1}{3} & \frac{1}{6} & -\frac{1}{3} & \frac{1}{6} \\ \frac{1}{3} & \frac{1}{6} & -\frac{1}{3} & \frac{1}{6} & -\frac{1}{3} \end{bmatrix} + \frac{1}{\delta}\begin{bmatrix} -\frac{32}{9} & \frac{13}{18} & \frac{13}{18} & \frac{13}{18} & \frac{13}{18} \\ \frac{13}{18} & -\frac{41}{36} & \frac{11}{18} & -\frac{23}{36} & \frac{11}{18} \\ \frac{13}{18} & \frac{11}{18} & -\frac{41}{36} & \frac{11}{18} & -\frac{23}{36} \\ \frac{13}{18} & -\frac{23}{36} & \frac{11}{18} & -\frac{41}{36} & \frac{11}{18} \\ \frac{13}{18} & \frac{11}{18} & -\frac{23}{36} & \frac{11}{18} & -\frac{41}{36} \end{bmatrix} +$$
$$+ \begin{bmatrix} \frac{2}{27} & -\frac{19}{216} & -\frac{19}{216} & -\frac{19}{216} & -\frac{19}{216} \\ -\frac{19}{216} & -\frac{25}{108} & \frac{31}{216} & \frac{29}{108} & \frac{31}{216} \\ -\frac{19}{216} & \frac{31}{216} & -\frac{25}{108} & \frac{31}{216} & \frac{29}{108} \\ -\frac{19}{216} & \frac{29}{108} & \frac{31}{216} & -\frac{25}{108} & \frac{31}{216} \\ -\frac{19}{216} & \frac{31}{216} & \frac{29}{108} & \frac{31}{216} & -\frac{25}{108} \end{bmatrix} + \dots . \tag{21}$$

Using the same procedure as in the 1D case, we obtain that RBF interpolation in the limit $\delta \to 0$ converges to the polynomial $p(x,y) = p_0(x,y) + \delta p_1(x,y) + \dots$, with

$$p_0(x,y) = \begin{bmatrix} 1-x^2-y^2 \\ \frac{y(y+1)}{2} \\ \frac{x(x+1)}{2} \\ \frac{y(y-1)}{2} \\ \frac{x(x-1)}{2} \end{bmatrix}^T \begin{bmatrix} f_{0,0} \\ f_{0,1} \\ f_{1,0} \\ f_{0,-1} \\ f_{-1,0} \end{bmatrix}, \tag{22}$$

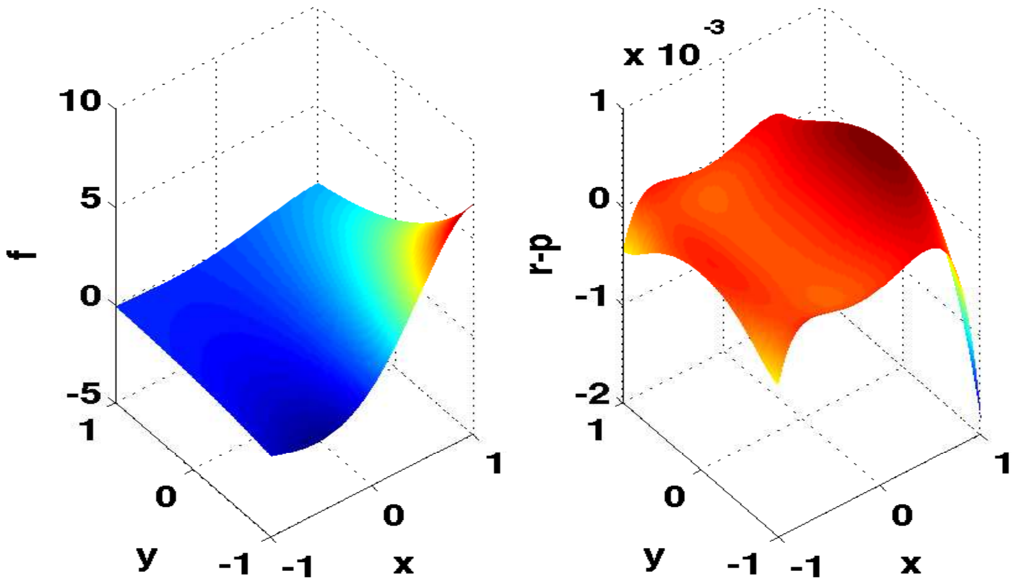

Fig. 1. Left: function (24). Right: difference between RBF interpolation for $\delta = 0.05$ using the five nodes stencil $[(0,0),(0,1),(1,0),(0,-1),(-1,0))]$, and the RBF interpolation in the limit $\delta \to 0$ ($p(x,y) = p_0(x,y) + \delta\, p_1(x,y) + \delta^2\, p_2(x,y)$ for $\delta = 0.05$).

and

$$p_1(x,y) = 12 \begin{bmatrix} 10x^4 + 20x^2y^2 - 10x^2 + 10y^4 - 10y^2 \\ 20x^4 - 5x^2y^2 - 3x^2y - 2x^2 - 7y^4 - 3y^3 + 7y^2 + 3y \\ -7x^4 - 3x^3 - 5x^2y^2 + 7x^2 - 3xy^2 + 3x + 2y^4 - 2y^2 \\ 20x^4 - 5x^2y^2 + 3x^2y - 2x^2 - 7y^4 + 3y^3 + 7y^2 - 3y \\ -7x^4 + 3x^3 - 5x^2y^2 + 7x^2 + 3xy^2 - 3x + 2y^4 - 2y^2 \end{bmatrix}^T \begin{bmatrix} f_{0,0} \\ f_{0,1} \\ f_{1,0} \\ f_{0,-1} \\ f_{-1,0} \end{bmatrix}. \quad (23)$$

In (22) and (23), $f_{0,0}$, $f_{0,1}$, $f_{1,0}$,$f_{0,-1}$, and $f_{-1,0}$, are the values of the function at the corresponding nodes. Similarly, $p_2(x,y)$ is a sixth degree polynomial, and so on.

For instance, we can use the five node stencil to interpolate the function

$$f(x,y) = e^{x-y} \sin(2x) \quad (24)$$

in a square domain. The left side of Figure 1 shows the function (24). The RBF interpolant is computed as

$$s(x,y;\delta) = \sum_{i=1}^{5} \alpha_i(\delta)\, \sqrt{1+\delta\,[(x-x_i)^2+(y-y_i)^2]}\,, \tag{25}$$

where the coefficients $\alpha_i(\delta)$, $i = 1, 2, \ldots, 5$, are the components of the vector

$$\boldsymbol{\alpha}(\delta) = A^{-1}(\delta)\,\mathbf{f}(x,y) \;=\; A^{-1}(\delta) \begin{bmatrix} 0 \\ 0 \\ e\sin(2) \\ 0 \\ e^{-1}\sin(-2) \end{bmatrix}. \tag{26}$$

Here, $A^{-1}(\delta)$ is the inverse matrix given in (21). We have also computed the interpolating polynomial in the limit $\delta \to 0$. The right side of Figure 1 shows the difference between the RBF interpolation $s(x,y;0.05)$ and the limit polynomial interpolant $p(x,y) = p_0(x,y) + 0.05\,p_1(x,y) + 0.05^2\,p_2(x,y)$. The error is zero at the interpolating points, and maximum at the nodes in the corners of the square. The difference between $s(x,y;0.05)$ and the interpolating polynomial decreases with increasing number of terms: $\parallel s - p_0 \parallel_\infty = 0.0691$, $\parallel s - p_0 - 0.05\,p_1 \parallel_\infty = 0.0105$, $\parallel s - p_0 - 0.05\,p_1 - 0.05^2\,p_2 \parallel_\infty = 0.0019$.

3.1 Limiting polynomial interpolant

In a seminal paper, Driscoll and Fornberg [6] studied RBF interpolation in the limit of increasingly flat basis functions. They showed that, *subject to some easily stated conditions on $\phi(r)$*, RBF interpolation in 1D converges to the Lagrange minimal-degree interpolating polynomial. They also found that the situation in 2D is more complicated. To this end, they used a symbolic language to compute the limiting polynomial interpolant for some simple illustrative examples. They showed that in some cases the limit does not exist, and in others the limit is a low degree polynomial. In cases when the limit exists, the interpolating polynomial might be different depending on the RBF function used. Other authors have also made significant contributions to the study of RBF interpolation in this limit [18,27,31].

In this Section, we use the Laurent series of the interpolation weights (5) to analyze some of the illustrative examples used by these authors. We will derive not only the leading order polynomial that approximates the RBF interpolant, but also the higher order polynomial terms.

Consider the case of 2D interpolation using multiquadrics as RBFs. We use the four interpolating nodes $[(0,0),(0,1),(1,0),(1,1)]$, as in Example 4 in [6].

Using the procedure described in Section 2, the leading polynomial approximation to the RBF interpolant is

$$p_0(x,y) = \begin{bmatrix} 1-x-y+x\,y \\ y\,(1-x) \\ x\,(1-y) \\ x\,y \end{bmatrix}^T \begin{bmatrix} f_{0,0} \\ f_{0,1} \\ f_{1,0} \\ f_{1,1} \end{bmatrix}.$$

In [6], this stencil was used to interpolate the function $f(x,y) = x-2y+3xy$, and therefore,

$$p_0(x,y) = \begin{bmatrix} 1-x-y+x\,y \\ y\,(1-x) \\ x\,(1-y) \\ x\,y \end{bmatrix}^T \begin{bmatrix} 0 \\ -2 \\ 1 \\ 2 \end{bmatrix} = x-2y+3xy\,.$$

In this case, $p_0(x,y)$ coincides with the function being interpolated and with the Lagrange interpolating polynomial. Exactly the same result is obtained (to leading order) if the inverse multiquadric RBF $\phi(r) = 1/\sqrt{1+\delta r^2}$ is used for interpolation. The polynomials $p_n(x,y)$ of higher order $(n>1)$ are, however, different depending on the used RBF.

Example 5 in [6] considers interpolation using the six nodes $[(0,0),(0,1),(1,0),(1,1),(0,0.5),(1,0.5)]$. With multiquadrics as RBF, the interpolant converges to

$$p_0(x,y) = \begin{bmatrix} -\frac{1}{2}x^3+\frac{5}{4}x^2-2xy^2+3xy-\frac{7}{4}x+2y^2-3y+1 \\ -\frac{1}{2}x^3+\frac{5}{4}x^2-2xy^2+xy-\frac{3}{4}x+2y^2-y \\ \frac{1}{2}x^3-\frac{1}{4}x^2+2xy^2-3xy+\frac{3}{4}x \\ \frac{1}{2}x^3-\frac{1}{4}x^2+2xy^2-xy-\frac{1}{4}x \\ x^3-\frac{5}{2}x^2+4xy^2-4xy+\frac{3}{2}x-4y^2+4y \\ -x^3+\frac{1}{2}x^2-4xy^2+4xy+\frac{1}{2}x \end{bmatrix}^T \begin{bmatrix} f_{0,0} \\ f_{0,1} \\ f_{1,0} \\ f_{1,1} \\ f_{0,0.5} \\ f_{1,0.5} \end{bmatrix}.$$

In [6], this stencil is used to interpolate $f(x,y) = x-y-2xy-2y^2$, and therefore,

$$p_0(x,y) = \begin{bmatrix} -\frac{1}{2}x^3 + \frac{5}{4}x^2 - 2xy^2 + 3xy - \frac{7}{4}x + 2y^2 - 3y + 1 \\ -\frac{1}{2}x^3 + \frac{5}{4}x^2 - 2xy^2 + xy - \frac{3}{4}x + 2y^2 - y \\ \frac{1}{2}x^3 - \frac{1}{4}x^2 + 2xy^2 - 3xy + \frac{3}{4}x \\ \frac{1}{2}x^3 - \frac{1}{4}x^2 + 2xy^2 - xy - \frac{1}{4}x \\ x^3 - \frac{5}{2}x^2 + 4xy^2 - 4xy + \frac{3}{2}x - 4y^2 + 4y \\ -x^3 + \frac{1}{2}x^2 - 4xy^2 + 4xy + \frac{1}{2}x \end{bmatrix}^T \begin{bmatrix} 0 \\ -3 \\ 1 \\ -4 \\ -1 \\ -1 \end{bmatrix} = \\ = 2x - y - 2xy - x^2 - 2y^2 \, . \quad (27)$$

A very complex algebraic equation was obtained in [6], instead of this low order polynomial.

If instead of using multiquadrics we use the inverse quadratic $\phi(r) = 1/(1+\delta r^2)$ as RBF, the leading order approximation to the RBF interpolant is given by

$$p_0(x,y) = \frac{1}{35}\begin{bmatrix} -10x^3 + 22x^2 - 70xy^2 + 105xy - 47x + 70y^2 - 105y + 35 \\ -10x^3 + 22x^2 - 70xy^2 + 35xy - 12x + 70y^2 - 35y \\ 10x^3 - 8x^2 + 70xy^2 - 105xy + 33x \\ 10x^3 - 8x^2 + 70xy^2 - 35xy - 2x \\ 20x^3 - 44x^2 + 140xy^2 - 140xy + 24x - 140y^2 + 140y \\ -20x^3 + 16x^2 - 140xy^2 + 140xy + 4x \end{bmatrix}^T \begin{bmatrix} 0 \\ -3 \\ 1 \\ -4 \\ -1 \\ -1 \end{bmatrix} = \\ = \frac{7}{5}x - y - \frac{2}{5}x^2 - 2xy - 2y^2 \, , \quad (28)$$

which coincides with that computed in [6] (there is an error in the sign of the xy term). Notice that, in this case, the leading order approximation to the RBF interpolant is different depending on the RBF used: equation (27) for multiquadrics, and equation (28) for inverse quadratic. In fact, if we use the inverse multiquadric we obtain to leading order $p_0(x,y) = \frac{3}{2}x - y - \frac{1}{2}x^2 - 2xy - 2y^2$, a third low order polynomial which also interpolates the function at the given nodes.

Example 6 in [6] considers interpolation in a unit square using four non-equispaced nodes $[(0,1),(1/4,1),(1/2,1/2),(1,3/4)]$. Again, in this case, the interpolating polynomial depends on the particular RBF used for interpolation. For instance, using multiquadrics the leading order interpolating polynomial is

$$
p_0(x,y) = \frac{1}{115}\begin{bmatrix} -420x^2 - 80xy - 485x + 180y^2 - 355y + 290 \\ -504x^2 + 96xy + 490x - 216y^2 + 610y - 394 \\ -84x^2 + 16xy + 5x - 36y^2 - 205y + 241 \\ 168x^2 - 32xy - 10x + 72y^2 - 50y - 22 \end{bmatrix}^T \begin{bmatrix} f_{0,1} \\ f_{1/4,1} \\ f_{1/2,1/2} \\ f_{1,3/4} \end{bmatrix}.
$$

In [6], this stencil is used to interpolate the function $f(x, y) = (x + 2y)/(3x - y + 2)$. Thus, the resulting interpolating polynomial is

$$
p_0(x,y) = \frac{846}{391}x^2 - \frac{1128}{2737}xy - \frac{2335}{782}x + \frac{829}{894}y^2 - \frac{457}{912}y + \frac{757}{481}, \tag{29}
$$

which coincides with that reported in [6]. Also for the inverse quadratic and inverse multiquadric we obtain identical results to those obtained in [6].

Example 7 in [6] considers interpolation in the unit square using 5×5 equispaced nodes. According to [6], the RBF interpolant diverges as $\delta \to 0$. We have not been able to verify this case because the singularity of the interpolation matrix is of order 7, which we can not compute in reasonable times.

Larsson and Fornberg [18] analyzed theoretically and numerically the behavior of multivariate interpolants in the limit $\delta \to 0$. They derived information about the degree of the limiting polynomial and gave precise conditions on the RBFs and the data points for different limiting results. In particular, they analyzed when the limiting polynomial is unique (independent of the used RBF) and when it is divergent. As an example of a divergent RBF, they considered the case of six equispaced nodes along a line. These nodes are $[(0, 0), (1/5, 1/5), (2/5, 2/5), (3/5, 3/5), (4/5, 4/5), (1, 1)]$. Following [18], we use cardinal data for the interpolant. That is, the interpolant takes the value one at the first node, and zero in the other five nodes. In the case of multiquadrics, the resulting polynomial is

$$
p_{-1}(x,y) = \frac{625}{29568}(x-y)^2(18 - 7x - 7y)/\delta, \tag{30}
$$

which coincides with that reported in [18]. The next higher order polynomial is

$$
\begin{aligned}
p_0(x,y) = &-\frac{1756}{1283}\left(x^5 + y^5\right) - \frac{2053}{444}\left(x^4y + xy^4\right) - \frac{5714}{813}\left(x^3y^2 + x^2y^3\right) + \\
&+\frac{3507}{496}\left(x^4 + y^4\right) + \frac{652}{32}\left(x^3y + xy^3\right) + \frac{2218}{89}x^2y^2 - \frac{7765}{553}\left(x^3 + y^3\right) - \\
&-\frac{9099}{301}\left(x^2y + xy^2\right) + \frac{45625}{3388}\left(x^2 + y^2\right) + \frac{4447}{223}xy - \frac{137}{24}(x+y) + 1 .
\end{aligned} \tag{31}
$$

This divergence is not generic to all RBFs. For instance, Fornberg, Wright and Larsson [13] proved that interpolation with nodes on a line always converges in the case of Gaussian RBF. In the following, we also observe that convergence depends on the number of nodes.

To understand this divergence, we consider the problem of interpolation using equally spaced nodes along the x-axis. For instance, using the three equispaced nodes $[(-1,0),(0,0),(1,0)]$, the interpolation matrix is identical to that for the 1D case (see Eq. (10)). The interpolation matrix only depends on the distance between nodes. Thus, the Laurent series of the inverse is given by (11), and the Taylor series of the weights by (12). The only difference with the 1D case comes from the power expansion of the radial basis functions (13). Hence, the value of the RBF interpolant at position $\mathbf{x}$, in the limit $\delta \to 0$, results in

$$\begin{aligned} \lim_{\delta\to 0} s(x;\delta) &= \lim_{\delta\to 0} \boldsymbol{\alpha}(\delta)^T \begin{bmatrix} 1+\delta\,\dfrac{(x+1)^2+y^2}{2} - \delta^2\,\dfrac{[(x+1)^2+y^2]^2}{8} + \dots \\ 1+\delta\,\dfrac{x^2+y^2}{2} - \delta^2\,\dfrac{[x^2+y^2]^2}{8} + \dots \\ 1+\delta\,\dfrac{(x-1)^2+y^2}{2} - \delta^2\,\dfrac{(x-1)^4}{8} + \dots \end{bmatrix} = \\ &= 1 - x^2 - \frac{y^2}{2}\,. \end{aligned} \tag{32}$$

The same procedure can be used to analyze the limiting 2D interpolant for a higher number of equispaced nodes along a line. In the case of four equispaced nodes $[(0,0),(1/3,0),(2/3,0),(1,0)]$ in the interval $[0,1]$, the limiting interpolating polynomial for cardinal data is

$$p_0(x,y) = -\frac{9}{2}x^3 + 9x^2 - \frac{9}{4}xy^2 - \frac{11}{2}x + \frac{9}{4}y^2 + 1\,. \tag{33}$$

Notice that setting $y=0$ in (32) and (33), we recover the Lagrange interpolating polynomial in 1D. For the case of five equispaced nodes $[(0,0),(1/4,0),(1/2,0),(3/4,0),(1,0)]$, the limiting interpolating polynomial for cardinal data is

$$p_{-1}(x,y) = \frac{32}{21}y^2\,, \tag{34}$$

$$p_0 = \frac{32}{3}x^4 - \frac{80}{3}x^3 + \frac{48}{7}x^2y^2 + \frac{70}{3}x^2 - \frac{200}{21}xy^2 - \frac{25}{3}x + \frac{650}{147}y^2 + 1\,. \tag{35}$$

The interpolant diverges for $y \neq 0$, but for $y = 0$ we still recover the 1D interpolating polynomial. Similarly, for the case of six equispaced nodes $[(0,0),(1/5,0),(2/5,0),(3/5,0),(4/5),(1,0)]$, the limiting interpolating polynomial for cardinal data is

$$p_{-1}(x,y) = \left[\frac{1875}{616}y^2 - \frac{625}{264}xy^2\right], \tag{36}$$

$$p_0 = -\frac{625}{24}x^5 + \frac{625}{8}x^4 - \frac{3125}{176}x^3y^2 - \frac{2125}{24}x^3 + \frac{8611}{246}x^2y^2 + \tag{37}$$

$$+\frac{375}{8}x^2 - \frac{11729}{493}xy^2 - \frac{137}{12}x + \frac{6649}{951}y^2 + 1, \tag{38}$$

which, again, diverges as δ^{-1}.

For the case of seven nodes the interpolating polynomial diverges as δ^{-2}. The limiting interpolating polynomial for cardinal data is

$$p_{-2}(x,y) = 3.38824\,y^2, \tag{39}$$

$$p_{-1}(x,y) = 8.13332\,y^2 - 9.2984\,x\,y^2 + 6.35294\,x^2\,y^2. \tag{40}$$

As the number of nodes increases the interpolant diverges faster.

It is interesting to see what happens in 3D. If we use nodes located along a line, the limiting interpolating polynomial in 3D coincides with that in 2D if y^2 is replaced by $y^2 + z^2$. For instance, in the case of four equispaced nodes $[(0,0,0), (1/3,0,0), (2/3,0,0), (1,0,0)]$, the limiting interpolating polynomial for cardinal data is

$$p_0(x,y) = -\frac{9}{2}x^3 + 9x^2 - \frac{9}{4}x(y^2+z^2) - \frac{11}{2}x + \frac{9}{4}(y^2+z^2) + 1. \tag{41}$$

Next, let us see what happens when the nodes in 3D are located on a plane. Using the five node stencil, $[(0,0,0), (1,0,0), (0,1,0), (-1,0,0), (0,-1,0)]$, the limiting interpolating polynomial is

$$p_0(x,y) = \begin{bmatrix} 1 - x^2 - y^2 - \frac{2}{3}z^2 \\ \frac{x(x+1)}{2} + \frac{1}{6}z^2 \\ \frac{y(y+1)}{2} + \frac{1}{6}z^2 \\ \frac{x(x-1)}{2} + \frac{1}{6}z^2 \\ \frac{y(y-1)}{2} + \frac{1}{6}z^2 \end{bmatrix}^T \begin{bmatrix} f_{0,0,0} \\ f_{1,0,0} \\ f_{0,1,0} \\ f_{-1,0,0} \\ f_{0,-1,0} \end{bmatrix}, \tag{42}$$

which coincides with (22) when $z = 0$, since in (22), we used the same layout for the nodes but in 2D. In the case of nine equispaced nodes $[(0,0,0), (1,0,0), (0,1,0), (-1,0,0), (0,-1,0), (1,1,0), (-1,1,0), (1,-1,0), (-1,-1,0)]$ located on the plane $z = 0$, the interpolant diverges as

$$p(x,y) \approx \frac{z^2}{\delta}\frac{1}{77}\begin{bmatrix} 4 \\ -2 \\ -2 \\ -2 \\ -2 \\ 2 \\ 2 \\ 2 \\ 2 \end{bmatrix}^T \begin{bmatrix} f_{0,0,0} \\ f_{1,0,0} \\ f_{0,1,0} \\ f_{-1,0,0} \\ f_{0,-1,0} \\ f_{1,1,0} \\ f_{-1,1,0} \\ f_{1,-1,0} \\ f_{-1,-1,0} \end{bmatrix} . \tag{43}$$

Another case analyzed in [18] is when the points lie on a circle. Consider the case in which the nodes are distributed as

$$\mathbf{x}_k = \frac{1}{2}\left[\cos\left(\frac{(k-1)\pi}{3}\right)+1, \sin\left(\frac{(k-1)\pi}{3}\right)+1\right], \quad k = 1,2,\ldots,K\,. \tag{44}$$

Buhmann and Dinew [4] proved that all RBF types give the same flat basis function limit (provided that the radial function's Taylor expansion has no zero coefficients, which is virtually always valid). For instance, consider the case $K = 6$ for which there is no unique interpolating polynomial. However, all RBFs have the same limit interpolant of degree 3. Using cardinal data ($f(x_1,y_1) = 1, f(x_k,y_k) = 0$ for $k = 2,\ldots,6$), the limit interpolant is

$$p_0(x,y) = \frac{1}{6}\left(8x^3 - 24xy^2 - 4x^2 + 4y^2 + 24xy - 4x - 4y + 1\right) . \tag{45}$$

A very interesting case is the seven node stencils (hexagonal stencils), since this is the preferred arrangement when one tries to distribute nodes uniformly on a domain [10]. This is similar to (44), but including the center node. Therefore, $\mathbf{x}_1 = (0,0)$, $\mathbf{x}_2 = (1,0)$, and $\mathbf{x}_k = (\cos\left(\frac{2(k-2)\pi}{6}\right), \sin\left(\frac{2(k-2)\pi}{6}\right))$ for $k = 3,\ldots,7$. In this case, the limit interpolant is

$$p_0(x,y) = \begin{bmatrix} 1 - x^2 - y^2 \\ \frac{1}{6}x^3 + \frac{1}{2}x^2 - \frac{1}{2}xy^2 + \frac{1}{3}x - \frac{1}{6}y^2 \\ -\frac{1}{6}x^3 + \frac{1}{2}xy^2 + \frac{1}{\sqrt{3}}xy + \frac{1}{6}x + \frac{1}{3}y^2 + \frac{1}{2\sqrt{3}}y \\ +\frac{1}{6}x^3 - \frac{1}{2}xy^2 - \frac{1}{\sqrt{3}}xy - \frac{1}{6}x + \frac{1}{3}y^2 + \frac{1}{2\sqrt{3}}y \\ -\frac{1}{6}x^3 + \frac{1}{2}x^2 + \frac{1}{2}xy^2 - \frac{1}{3}x - \frac{1}{6}y^2 \\ +\frac{1}{6}x^3 - \frac{1}{2}xy^2 + \frac{1}{\sqrt{3}}xy - \frac{1}{6}x + \frac{1}{3}y^2 - \frac{1}{2\sqrt{3}}y \\ -\frac{1}{6}x^3 + \frac{1}{2}xy^2 - \frac{1}{\sqrt{3}}xy + \frac{1}{6}x + \frac{1}{3}y^2 - \frac{1}{2\sqrt{3}}y \end{bmatrix}^T \begin{bmatrix} f_0 \\ f_1 \\ f_2 \\ f_3 \\ f_4 \\ f_5 \\ f_6 \end{bmatrix}, \quad (46)$$

p_1 is a polynomial of degree 5, and p_2 a polynomial of degree 7.

3.2 Optimal shape parameter

In 1979, R. Franke published a report [17] in which he compared the performance of more than thirty algorithms for scattered data interpolation. He used a set of six 2D test functions defined in the unit square (see Appendix B), and three data sets composed of 25, 33, and 100 scattered interpolation points. The RBF multiquadric method performed exceptionally well and, as a result, the multiquadric method began to generate considerable interest among researchers. Some years later, Carlson and Foley [5] used the same set of functions and interpolation nodes (adding some additional sets of interpolation nodes) to analyze the dependence of the interpolation accuracy on the multiquadric shape parameter, and they proposed a simple algorithm to compute its optimal value.

To evaluate the accuracy for each value of the shape parameter δ, they computed the RBF interpolant (1) at a set of 1089 (33×33) equispaced nodes, and defined the root-mean-square (RMS) error as

$$\mathrm{RMS}(\delta) = \sqrt{\frac{1}{1089}\sum_{k=1}^{1089}\left[f_n(x_k,y_k) - s(x_k,y_k;\delta)\right]^2}, \quad n = 1, 2, \ldots, 6, \quad (47)$$

where $f_n(x,y)$ is one of the six test functions used by Franke.

Figure 2 summarizes the results in four different interpolation data sets: Franke's 25, 33 and 100 nodes, and Carlson and Foley's 58 nodes (union of the 25 and 33 data sets). For ease of comparison with the results in [5], we use $c^2 = h^2/\delta$ as shape parameter, and write the multiquadric as $\phi_i(x,y) =$

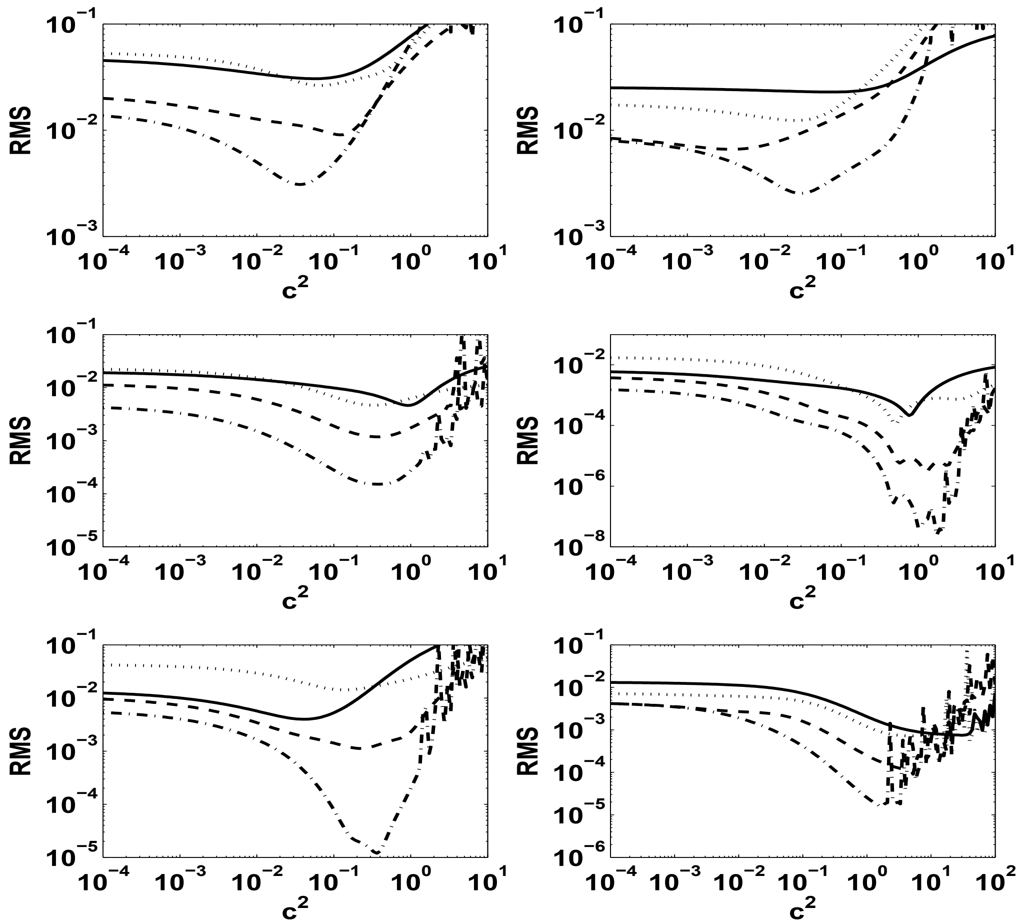

Fig. 2. RMS error versus c^2 for multiquadric interpolation to test functions f_1, f_2, $\dots f_n$ (from left to right and from top to bottom). Solid line: 25 interpolation nodes. Dotted: 33 interpolation nodes. Dashed: 58 interpolation nodes. Dot-dashed: 100 interpolation nodes.

$\sqrt{c^2 + (x - x_i)^2 + (y - y_i)^2}$. From these results, the authors concluded that the optimal shape parameter is essentially independent of both, the number and the location of the interpolation points. On the other hand, the optimal shape parameter strongly depends on the function being interpolated. It should also be pointed out that, for relatively large data sets (58 and 100 nodes) the numerical computation of the solution becomes ill-conditioned for values of c^2 slightly larger than one and, therefore, large round-off errors appear. In fact, very often, the optimal shape parameter is not determined by the minimal interpolation error but by the value of the shape parameter where ill-conditioning starts. This was not clearly observed in the results reported in [5] because the resolution in c^2 was rather small.

On the very crucial question of how to select an optimal value of the shape parameter, there have been a significant number of relevant contributions. Carlson and Foley [5] were the first to propose an *ad-hoc* method for selecting the optimal value. It was based on their experimental observations. Rippa [24] was the first to propose an algorithm based on the idea of *leave-one-out* cross validation (LOOCV), which is used in the statistics literature for a variety of parameter identification problems. Their method is based on minimizing a cost function that estimates the RMS error. To compute the cost function, they split off one single data point (x_k, y_k) at a time, calculate the approximation error $E_k = f_k - s_k$ of the interpolant at the removed node, and estimate a cost function by taking a norm of the vector $\mathbf{E} = [E_1, E_2, \ldots, E_n]^T$. The cost function depends on the shape parameter, and minimizing it yields the optimal value. The significant impact of Rippa's method in RBF research has led to a significant amount of work based on similar ideas [8,25,26,29,30].

We propose to use the polynomial approximation to the RBF interpolant in the limit of increasingly flat basis functions to compute the optimal value of the shape parameter. For the procedure to work, it is necessary that the interpolation error has a minimum in the region $\delta < 1$, where the polynomial approximation to the interpolant is valid.

We define the interpolation error at location $\mathbf{x}$ as $E(\mathbf{x}; \delta) = f(\mathbf{x}) - s(\mathbf{x}; \delta)$, where $f(\mathbf{x})$ and $s(\mathbf{x}; \delta)$ are the exact value of the function, and the RBF interpolant at $\mathbf{x}$, respectively. Our objective is to find the value of the shape parameter that minimizes some norm of $E(\mathbf{x}; \delta)$. Most methods use nonlinear minimization to find this optimal value, and this requires to solve interpolation equations several times. Instead, we use the power series approximation to the RBF interpolant (7) to estimate the optimal value of the shape parameter. To this end, we select a set $\mathbf{x}_k$, $k = 1, \ldots, M$, of M evaluation nodes, and compute the optimal value of δ by minimizing the error function

$$\mathcal{E}(\delta) = \| \sum_{k=1}^{M} (f(\mathbf{x}_k) - p_0(\mathbf{x}_k) - \delta\, p_1(\mathbf{x}_k) - \delta^2\, p_2(\mathbf{x}_k) - \ldots) \| , \tag{48}$$

where $\| \cdot \|$ denotes some norm. To minimize the infinite norm we use Matlab `fminimax` function. To minimize the two norm (to leading order) we compute

$$\delta_2^* = \mathtt{dot(f - p0, p1)/dot(p1, p1)}. \tag{49}$$

To check the method, we interpolate the function

$$f(x, y) = \sqrt{1 + 0.04\,((x - 1/2)^2 + (y - 1)^2)} \tag{50}$$

using the nodes $[(1/2, 1/2), (1/2, 1), (1, 1/2), (1/2, 0), (0, 1/2), (1, 1), (1, 0), (0, 0),$

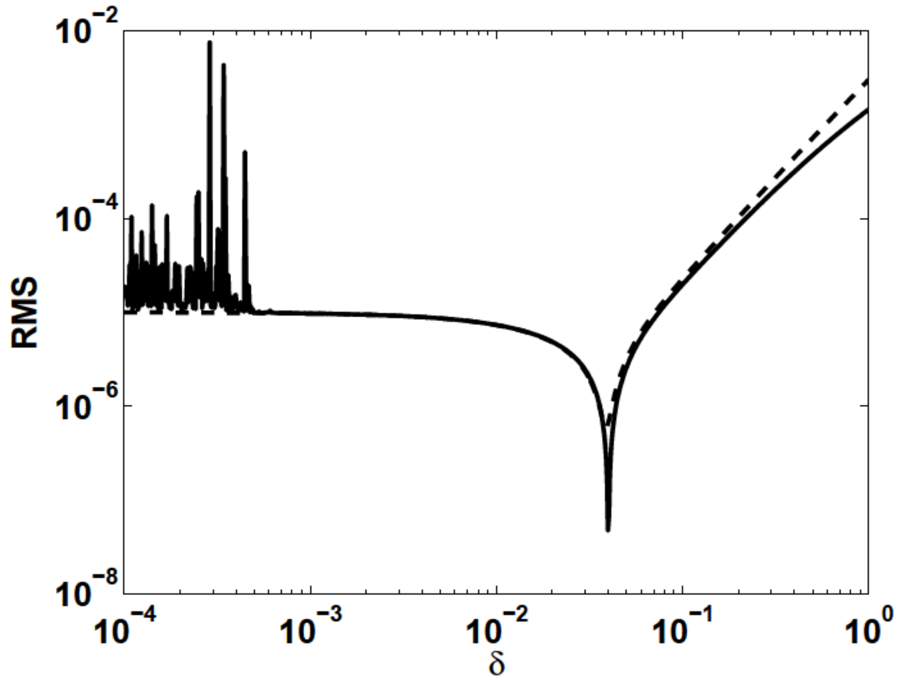

Fig. 3. Solid line: RMS error of the RBF interpolant for function (50). Dashed line: polynomial approximation $p(x, y) = p_0(x, y) + \delta p_1(x, y) + \delta^2 p_2(x, y)$.

$(0, 1)]$. The leading order $p_0(x, y)$ of the approximation polynomial of the RBF interpolant is written in Appendix A.

The solid line in Figure 3 shows the RMS error in a grid of 33×33 evaluation points ($N = 1089$) as a function of δ. With a dashed line, we show the RMS error using the second order polynomial approximation to the RBF interpolant, i.e., $p(x, y) = p_0(x, y) + \delta p_1(x, y) + \delta^2 p_2(x, y)$. Notice that there is a minimum near $\delta = 0.04$ which is also well defined using the polynomial approximation. In fact, minimizing the infinite norm using yields $\delta^*_\infty = 0.0388$, and minimizing the two norm (to leading order) yields $\delta^*_2 = 0.0303$.

Figure 4 shows the same analysis for the function

$$f(x, y) = \sqrt{1 + 2\,10^{-4}\left((x - 1/2)^2 + (y - 1)^2\right)}\,. \tag{51}$$

In this case, the optimal shape parameter lies in the region of ill-conditioning and, therefore, it is difficult to compute the optimal value of δ numerically. However, it is straightforward to compute it using the polynomial approximation. Using l_∞-norm minimization yields $\delta^*_\infty = 1.982\,10^{-4}$, and using l_2-norm minimization $\delta^*_2 = 1.545\,10^{-4}$.

Larsson and Fornberg [18] studied the error dependence with the shape parameter for the function

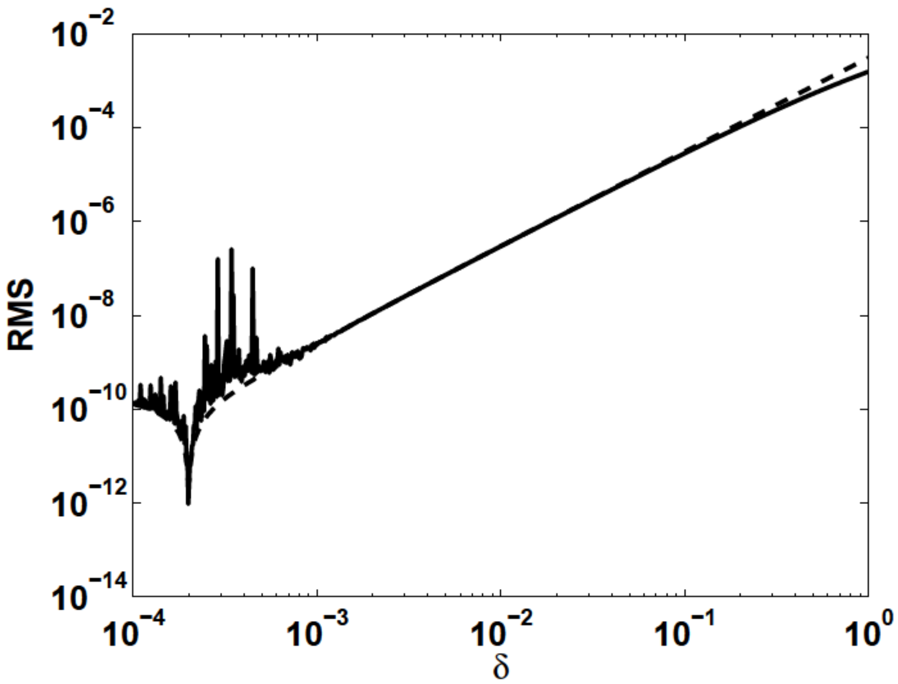

Fig. 4. Solid line: RMS error of the RBF interpolant for function (51). Dashed line: polynomial approximation $p(x, y) = p_0(x, y) + \delta p_1(x, y) + \delta^2 p_2(x, y)$.

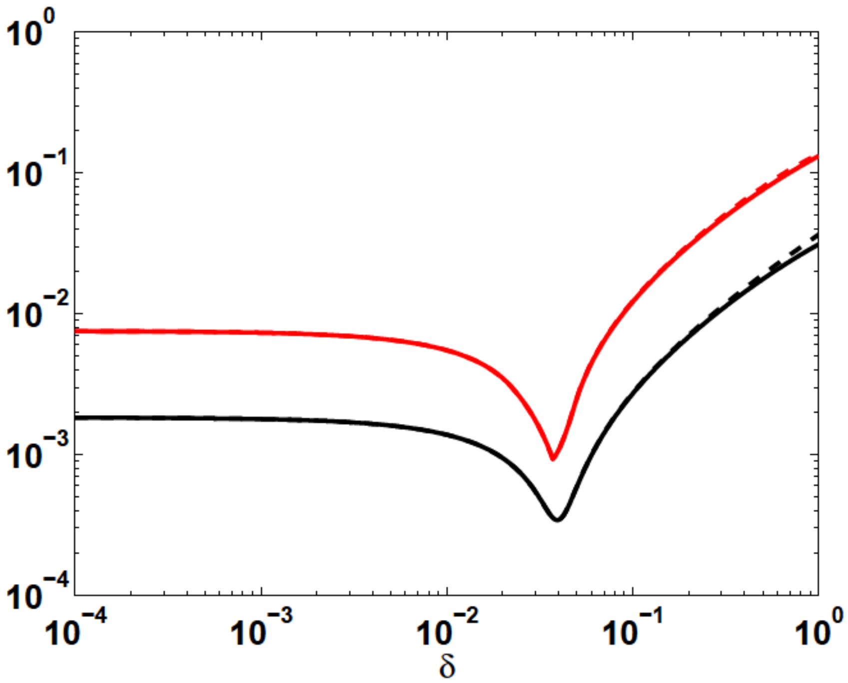

Fig. 5. Solid line: RMS error (black) and maximum error (red) of the RBF interpolant for function (52). Dashed line: polynomial approximation $p(x, y) = p_0(x, y) + \delta p_1(x, y) + \delta^2 p_2(x, y)$.

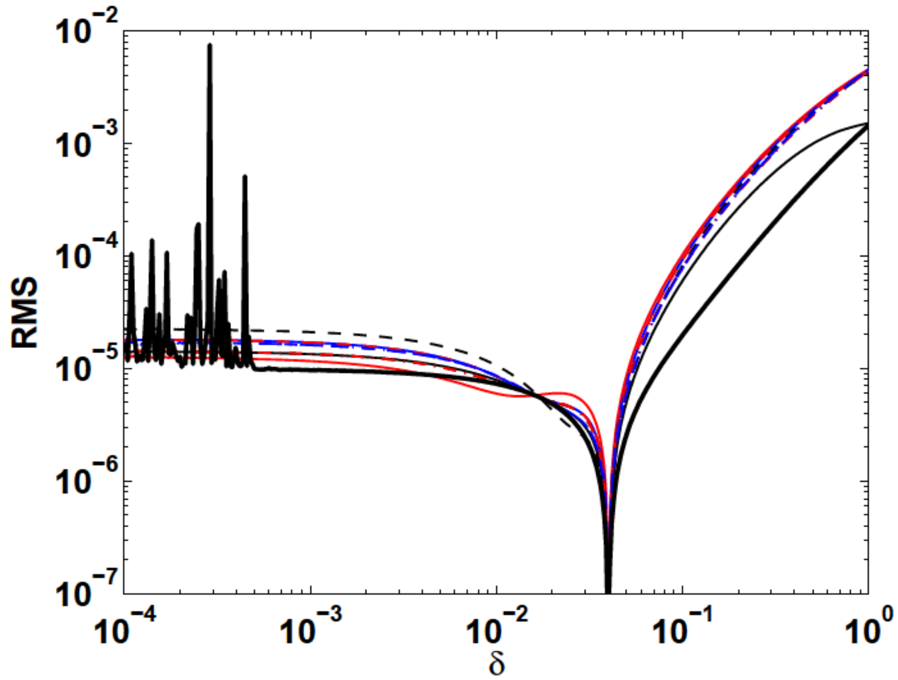

Fig. 6. Thin lines: RMS error of the RBF interpolant for function (50) using eight nodes out of nine $([(1/2,1/2),(1/2,1),(1,1/2),(1/2,0),(0,1/2),(1,1),(1,0),(0,0),(0,1)])$. Thick line: RMS error of the RBF interpolant for function (50) using these nine nodes.

$$f(x,y) = \frac{25}{25+(x-1/5)^2+2y^2}, \tag{52}$$

using nodes $[(1/10,4/5),(1/5,1/5),(3/10,1),(3/5,1/2),(4/5,3/5),(1,1)]$. Figure 5 shows the RMS error and the maximum error ($\| E \|_\infty$) as a function of δ for the RBF interpolant (solid line) and for the polynomial approximation $p(x,y) = p_0(x,y) + \delta p_1(x,y) + \delta^2 p_2(x,y)$ (dashed line). The error is computed in the fine grid of 33×33 equispaced nodes. Using l_∞-norm minimization yields $\delta^*_\infty = 0.0371$, and using l_2-norm minimization $\delta^*_2 = 0.0389$.

The method just described is effective when the interpolated function is known explicitly and, therefore, the errors can be computed exactly at each evaluation node. If the function is known only at the interpolation nodes, we can use a variation of Rippa's method but using the polynomial interpolation that approximates the RBF interpolant in the limit $\delta \to 0$. For instance, in the previous example of RBF interpolation using nine nodes, we can compute the RBF interpolant for each of the nine eight node stencils obtained by removing one node at a time, evaluate the error at the removed node as a function of δ, and compute the value of δ that minimizes some norm of the error.

Figure 6 shows the RMS error of the RBF interpolation of function (50) using eight nodes out of nine. We also show the RMS error using nine nodes. Notice

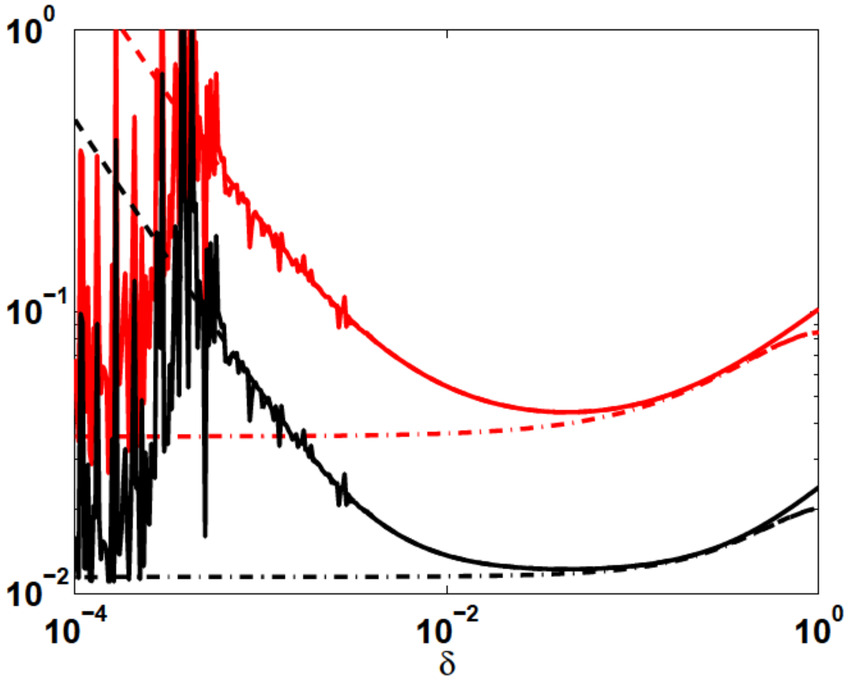

Fig. 7. Solid line: RMS error (black) and maximum error (red) of the RBF interpolant for function (52). Dashed line: polynomial approximation $p(x,y) = p_{-1}(x,y)/\delta + p_0(x,y) + \delta p_1(x,y) + \delta^2 p_2(x,y)$. Dot-dashed line: polynomial approximation $p(x,y) = p_0(x,y) + \delta p_1(x,y) + \delta^2 p_2(x,y)$.

that, although the RMS errors are different for each set of eight nodes, the location of the minimum error does not change. Thus, instead of having to compute the polynomial interpolants for each set of eight nodes, it suffices to compute just one eight node stencil to find the optimal value of δ. This is in agreement with the observation of Carlson and Foley [5] that the optimal shape parameter is quite independent on the number and positions of the interpolation nodes.

Thus, the method we propose to choose a *good* value of the shape parameter in RBF interpolation using n interpolation nodes, is the following:

- Select one subset of $n-1$ nodes out of the n nodes.
- Compute the polynomial interpolant (7) which approximates the $(n-1)$ RBF interpolant.
- Evaluate the error at the removed node $\mathbf{x}_i$: $E(\delta) = y_i - p_0(\mathbf{x}_i) - \delta p_1(\mathbf{x}_i) - \delta^2 p_1(\mathbf{x}_i)$.
- Find a *good* value of δ by solving the second order equation $y_i = p_0(\mathbf{x}_i) + \delta p_1(\mathbf{x}_i) + \delta^2 p_1(\mathbf{x}_i)$. If the computed value is negative take $\delta = 0$.
- Use the *good* value of δ to compute the RBF interpolant in n nodes.

As a last example, let us consider one case in which the RBF interpolant diverges as $\delta \to 0$. Using nodes on a line ([(0,0), (1/5,1/5), (2/5,2/5), (3/5,3/5),

$(4/5, 4/5), (1, 1)]$), the RBF interpolant diverges, and the limiting interpolant goes as

$$p(x, y) = \delta^{-1} p_{-1}(x, y) + p_0(x, y) + \delta p_1(x, y) + \delta^2 p_2(x, y) + \dots .$$

For function (52), Figure 7 shows the RMS error and the maximum error ($\| E \|_\infty$) as a function of δ for the RBF interpolant (solid line) and for the polynomial approximation $p(x, y) = p_{-1}(x, y)/\delta + p_0(x, y) + \delta p_1(x, y) + \delta^2 p_2(x, y)$ (dashed line). The error is computed in the fine grid of 33×33 equispaced nodes. Also shown (dot-dashed) is the polynomial interpolant excluding the divergent term. This is in agreement with the remark in page 11 of [18]: *if we could discard the divergent terms, we would get a limit that makes sense also with the non-unisolvent point sets.*

3.3 RBF Finite Difference Formulas (RBF-FD)

The power series expansion of the RBF interpolant in the limit $\delta \to 0$ (7) can also be used to derive RBF-FD formulas. It should be mentioned that other alternatives have been successfully proposed in the past to compute interpolants and RBF-FD weights in the limit of infinitely flat basis functions: Contour-Padé [12], RBF-QR [14,15] and RBF-GA [16].

Consider, for instance, the case of three equispaced nodes in 1D. To compute the first derivative of $f(x)$ at $x = 0$, we can just compute the first derivative of $p_0(x)$ in (17) at $x = 0$. This results in

$$h \frac{df}{dx} = \frac{1}{2} (f_1 - f_{-1}) .$$

The second order approximation is obtained by computing the first derivative of $p_1(x)$ and $p_2(x)$ in (18)-(19) at $x = 0$. Thus,

$$h \frac{df}{dx} = \frac{1}{2} (f_1 - f_{-1}) + \frac{\delta}{4} (f_1 - f_{-1}) - \frac{9\,\delta^2}{16} (f_1 - f_{-1}) + \dots .$$

Analogously, to compute the weights of the RBF-FD formula for the second derivative, we can just compute the second derivative of $p_0(x), p_1(x), \dots$ at $x = 0$. This results in

$$h^2 \frac{d^2 f}{dx^2} = (f_1 - 2f_0 + f_{-1}) + \delta (f_1 - 2f_0 + f_{-1}) - \delta^2 \left(\frac{15}{8} f_1 - 3f_0 + \frac{15}{8} f_{-1} \right) + \dots .$$

To compute the Laplacian of $f(\mathbf{x})$ at $\mathbf{x} = (0,0)$ using five equispaced nodes in 2D, we can just compute the Laplacian of $p_0(\mathbf{x})$ in (22) at $\mathbf{x} = (0,0)$. Thus,

$$h^2 \Delta f = -4f_{0,0} + f_{1,0} + f_{-1,0} + f_{0,1} + f_{0,-1} \, .$$

The next order approximation to the Laplacian is obtained by taking the Laplacian of $p_1(\mathbf{x})$ in (23) at $\mathbf{x} = (0,0)$, so

$$\begin{aligned} h^2 \Delta f &= -4f_{0,0} + f_{1,0} + f_{-1,0} + f_{0,1} + f_{0,-1} + \\ &+\delta \left[-\frac{10}{3} f_{0,0} + \frac{5}{6} \left(f_{1,0} + f_{-1,0} + f_{0,1} + f_{0,-1} \right) \right] . \end{aligned}$$

This formula is in agreement with that derived in [1].

The same method can be used to determine the weights of the RBF-FD formula for the nine-nodes Laplacian in the limit $\delta \to 0$. The result is

$$\begin{aligned} h^2 \Delta f = & -\frac{372}{77} f_{0,0} + \frac{109}{77} \left(f_{1,0} + f_{-1,0} + f_{0,1} + f_{0,-1} \right) - \\ & -\frac{16}{77} \left(f_{1,1} + f_{1,-1} + f_{-1,-1} + f_{-1,1} \right) + \qquad (53) \\ & +\delta \left[-\frac{1217}{205} f_{0,0} + \frac{2717}{1095} \left(f_{1,0} + f_{-1,0} + f_{0,1} + f_{0,-1} \right) - \right. \\ & \left. - \frac{1391}{1395} \left(f_{1,1} + f_{1,-1} + f_{-1,-1} + f_{-1,1} \right) \right] + \dots \qquad (54) \end{aligned}$$

To compute the local truncation error for this formula we expand $f_{i,j}$ around $f_{0,0}$ in a Taylor series leading to

$$\begin{aligned} \epsilon_9 \approx & \left[\frac{1}{12} \left(u^{(4,0)} + u^{(0,4)} \right) - \frac{16}{77} u^{(2,2)} \right] h^2 + \\ & + \frac{75}{154} \left(u^{(2,0)} + u^{(0,2)} \right) \delta - \frac{15}{22} \frac{\delta^2}{h^2} u \, , \end{aligned} \qquad (55)$$

where all derivatives are evaluated at $(0,0)$. Approximation (55) is in very good agreement with that computed numerically in [1]. In fact, the coefficients of $u^{(4,0)}$ and $u^{(0,4)}$ are identical, while the other coefficients are very close to those found in [1]. The coefficient of $u^{(2,2)}$ is $16/77 = 0.2078$ (instead of 0.2 in [1]), the coefficients of $u^{(2,0)}$ and $u^{(0,2)}$ are 75/154=0.487 (instead of 0.47 in [1]), and the coefficient of u is 15/22=0.6818 (instead of 2/3=0.6666 in [1]).

Notice that the leading order of the RBF-FD (53) formula does not coincide with the usual 9-point Laplacian [22]

$$h^2\,\Delta f = \frac{1}{6}\left[-20 f_{0,0} + 4\left(f_{1,0} + f_{-1,0} + f_{0,1} + f_{0,-1}\right) - \right.$$
$$\left. - \left(f_{1,1} + f_{1,-1} + f_{-1,-1} + f_{-1,1}\right)\right] .$$

This is due to the fact that the polynomial interpolant of degree two using nine nodes is not unique and, in fact, the polynomial which is obtained in the limit $\delta \to 0$ depends on the RBF used. In the case of the inverse multiquadric the limiting interpolant is

$$\begin{aligned} h^2\,\Delta f = &-\frac{1108}{243} f_{0,0} + \frac{311}{243}\left(f_{1,0} + f_{-1,0} + f_{0,1} + f_{0,-1}\right) - \\ &-\frac{34}{243}\left(f_{1,1} + f_{1,-1} + f_{-1,-1} + f_{-1,1}\right) + \qquad (56) \\ &+\delta\left[-\frac{1321}{137} f_{0,0} + \frac{13093}{3957}\left(f_{1,0} + f_{-1,0} + f_{0,1} + f_{0,-1}\right) - \right. \\ &\left. - \frac{1121}{1248}\left(f_{1,1} + f_{1,-1} + f_{-1,-1} + f_{-1,1}\right)\right] + \ldots , \qquad (57) \end{aligned}$$

and the corresponding local truncation error is

$$\begin{aligned} \epsilon_9 \approx &\left[\frac{1}{12}\left(u^{(4,0)} + u^{(0,4)}\right) - \frac{34}{243}\, u^{(2,2)}\right] h^2 + \\ &+ \frac{245}{162}\left(u^{(2,0)} + u^{(0,2)}\right)\delta + \frac{35}{18}\frac{\delta^2}{h^2}\, u . \qquad (58) \end{aligned}$$

The same method can be used to derive the weights of the RBF-FD thirteen-points formula for the Laplacian in the limit $\delta \to 0$. To order δ, the result is

$$\begin{aligned} h^2\,\Delta f = &-5 f_{0,0} + \frac{4}{3}\left(f_{1,0} + f_{-1,0} + f_{0,1} + f_{0,-1}\right) - \\ &-\frac{1}{12}\left(f_{2,0} + f_{-2,0} + f_{0,-2} + f_{0,2}\right) + \qquad (59) \\ &+\delta\left[-\frac{1502}{173} f_{0,0} + \frac{1318}{519}\left(f_{1,0} + f_{-1,0} + f_{0,1} + f_{0,-1}\right) - \right. \\ &-\frac{92}{173}\left(f_{1,1} + f_{1,-1} + f_{-1,-1} + f_{-1,1}\right) - \\ &\left. -\frac{935}{1038}\left(f_{2,0} + f_{-2,0} + f_{0,-2} + f_{0,2}\right)\right] + \ldots . \qquad (60) \end{aligned}$$

The corresponding local truncation error is

$$\epsilon_{13} \approx -\frac{1}{90}\left[u^{(6,0)}+u^{(0,6)}\right] h^4 - \left[\frac{935}{1038}\left(u^{(4,0)}+u^{(0,4)}\right) - \frac{92}{173}\, u^{(2,2)}\right] \delta\, h^2 + \\ + \left\{\frac{241}{92}\left[u^{(4,0)}+u^{(0,4)}\right] - \frac{2430}{337}\, u^{(2,2)} - \frac{1599}{346}\left[u^{(2,0)} + u^{(0,2)}\right]\right\} \delta^2 + \dots, \quad (61)$$

where all derivatives are evaluated at $(0,0)$. This formula is in close agreement with that derived numerically in [1].

In the case of seven nodes hexagonal stencils ($\mathbf{x}_1 = (0,0)$, $\mathbf{x}_2 = (1,0)$, and $\mathbf{x}_k = (\cos\left(\frac{2(k-2)\pi}{6}\right), \sin\left(\frac{2(k-2)\pi}{6}\right))$ for $k = 3, \dots, 7$), we can use p_0, p_1 and p_2 to compute the value of the weights of the RBF-FD formula for the Laplacian in the limit $\delta \to 0$. The resulting weights are

$$\alpha_0 = -\frac{4}{h^2} - \frac{\delta}{h^2}\frac{10}{3} + \frac{\delta^2}{h^2}\frac{157}{36},$$
$$\alpha_i = \frac{1}{h^2}\frac{2}{3} + \frac{\delta}{h^2}\frac{5}{9} - \frac{\delta^2}{h^2}\frac{199}{216}, \qquad i = 1, \dots, 6.$$

In the general case of a regular n node stencil distributed along the unit circle

$$x_i = \cos\left(\frac{2\pi(i-1)}{n-1}\right), \quad y_i = \sin\left(\frac{2\pi(i-1)}{n-1}\right), \qquad i = 1, \dots n-1$$

plus the central node $x_0 = y_0 = 0$, it is easy to prove that the following relationships hold

$$\sum_{i=1}^{n-1} x_i^{2m+1} = \sum_{i=1}^{n-1} y_i^{2m+1} = 0, \qquad i = 0, 1, \dots$$
$$\sum_{i=1}^{n-1} x_i^2 = \sum_{i=1}^{n-1} y_i^2 = \frac{n-1}{2}, \qquad \sum_{i=1}^{n-1} x_i^4 = \sum_{i=1}^{n-1} y_i^4 = \frac{6(n-1)}{16}.$$

Expanding $\sum_{i=0}^{n-1} \alpha_i\, u(x_i, y_i)$ in Taylor series around $(0,0)$, and collecting like powers of $\Delta x, \Delta y$, the leading order terms of the RBF-FD weights are

$$\alpha_0 = -\frac{4}{h^2} - \frac{\delta}{h^2}\frac{10}{3},$$
$$\alpha_i = \frac{1}{h^2}\frac{4}{n-1} + \frac{\delta}{h^2}\frac{10}{3\,(n-1)}, \qquad i = 1, \dots, n-1.$$

The corresponding local truncation error is

$$\epsilon_n(\mathbf{x}) = \frac{h^2}{16}\left[u^{(4,0)} + u^{(0,4)} + 2\, u^{(2,0)}\, u^{(0,2)}\right] + \frac{5\,\delta}{6}\left[u^{(2,0)} + u^{(0,2)}\right] + \dots.$$

Notice that this error is independent of n.

4 Conclusions

In this paper we propose a new approach to study RBF interpolation in the limit of flat RBFs. The method is based on the computation of the Laurent series of the inverse of the RBF interpolation matrix in powers of the parameter $\delta = (\epsilon h)^2$, where ϵ is the shape parameter and h is the characteristic internodal distance. The matrix coefficients of this Laurent series are computed using a semi-analytical algorithm described in [3]. It is semi-analytical because the Laurent series is derived from analytical formulas, but the implementation requires a significant amount of numerical computations.

We show that the Laurent series for the inverse of the interpolation matrix can be effectively used for different purposes: for the derivation of the limiting interpolation polynomial, for the computation of the optimal value of the shape parameter, and for the derivation of the weights of RBF-FD formulas and their corresponding local truncation error. An important advantage of our approach is that all the results obtained are extremely accurate, since the proposed procedure avoids the effects of ill-conditioning and the corresponding rounding errors.

It should be mentioned that the computational cost of the algorithm that provides the Laurent series of the inverse of the RBF interpolation matrix grows exponentially with the order of its singularity. Hence, although the algorithm is more efficient than symbolic languages, it is only feasible for singularities of order seven or less. As the order of the singularity increases with the number of interpolant nodes, it is only possible to obtain the Laurent series of the inverse for 36 nonequispaced nodes or 24 equispaced nodes in 2D, and for 84 nonequispaced nodes or 31 equispaced nodes in 3D.

5 Acknowledgments

This work has been supported by Spanish MICINN Grants FIS2010-18473, FIS2013-41802-R, CSD2010-00011.

A Limit polynomial interpolant

In the case of nine equispaced nodes the polynomial interpolant in the limit $\delta \to 0$ is

$$
\begin{aligned}
p_0(x,y) = \Big[&\frac{256}{77}x^4 - \frac{512}{77}x^3 + 16x^2y^2 - 16x^2y + \frac{320}{77}x^2 - 16xy^2 + \\
&+16xy - \frac{64}{77}x + \frac{256}{77}y^4 - \frac{512}{77}y^3 + \frac{320}{77}y^2 - \frac{64}{77}y\Big]\, f_{1/2,1/2} + \\
&+\Big[-\frac{128}{77}x^4 + \frac{256}{77}x^3 - 8x^2y^2 + 4x^2y - \frac{160}{77}x^2 + 8xy^2 - \\
&-4xy + \frac{32}{77}x - \frac{128}{77}y^4 + \frac{179}{77}y^3 - \frac{89}{154}y^2 - \frac{13}{154}y\Big]\, f_{1/2,1} + \\
&+\Big[-\frac{128}{77}x^4 + \frac{179}{77}x^3 - 8x^2y^2 + 8x^2y - \frac{89}{154}x^2 + 4xy^2 - \\
&-4xy + \frac{13}{154}x - \frac{128}{77}y^4 + \frac{256}{77}y^3 - \frac{160}{77}y^2 + \frac{32}{77}y\Big]\, f_{1,1/2} + \\
&+\Big[-\frac{128}{77}x^4 + \frac{256}{77}x^3 - 8x^2y^2 + 12x^2y - \frac{468}{77}x^2 + 8xy^2 - \\
&-12xy + \frac{340}{77}x - \frac{128}{77}y^4 + \frac{333}{77}y^3 - \frac{551}{154}y^2 + \frac{141}{154}y\Big]\, f_{1/2,0} + \\
&+\Big[-\frac{128}{77}x^4 + \frac{333}{77}x^3 - 8x^2y^2 + 8x^2y - \frac{551}{154}x^2 + 12xy^2 - \\
&-12xy + \frac{141}{154}x - \frac{128}{77}y^4 + \frac{256}{77}y^3 - \frac{468}{77}y^2 + \frac{340}{77}y\Big]\, f_{0,1/2} + \\
&+\Big[\frac{64}{77}x^4 - \frac{179}{154}x^3 + 4x^2y^2 - 2x^2y + \frac{89}{308}x^2 - 2xy^2 + \\
&+xy + \frac{13}{308}x + \frac{64}{77}y^4 - \frac{179}{154}y^3 + \frac{89}{308}y^2 + \frac{13}{308}y\Big]\, f_{1,1} + \\
&+\Big[\frac{64}{77}x^4 - \frac{179}{154}x^3 + 4x^2y^2 - 6x^2y + \frac{705}{308}x^2 - 2xy^2 + \\
&+3xy - \frac{295}{308}x + \frac{64}{77}y^4 - \frac{333}{154}y^3 + \frac{551}{308}y^2 - \frac{141}{308}y\Big]\, f_{1,0} + \\
&+\Big[\frac{64}{77}x^4 - \frac{333}{154}x^3 + 4x^2y^2 - 6x^2y + \frac{1167}{308}x^2 - 6xy^2 + \\
&+9xy - \frac{1065}{308}x + \frac{64}{77}y^4 - \frac{333}{154}y^3 + \frac{1167}{308}y^2 - \frac{1065}{308}y\Big]\, f_{0,0} + \\
&+\Big[\frac{64}{77}x^4 - \frac{333}{154}x^3 + 4x^2y^2 - 2x^2y + \frac{551}{308}x^2 - 6xy^2 + \\
&+3xy - \frac{141}{308}x + \frac{64}{77}y^4 - \frac{179}{154}y^3 + \frac{705}{308}y^2 - \frac{295}{308}y\Big]\, f_{0,1}
\end{aligned}
$$

B Functions

$$f_1(x,y) = \frac{3}{4}\exp\left[-\frac{(9x-2)^2+(9y-2)^2}{4}\right] + \frac{3}{4}\exp\left[-\frac{(9x+1)^2}{49} - \frac{9y+1}{10}\right] +$$
$$+\frac{1}{2}\exp\left[-\frac{(9x-7)^2+(9y-3)^2}{4}\right] - \frac{2}{10}\exp\left[-(9x-4)^2-(9y-7)^2\right]$$
$$f_2(x,y) = \frac{1}{9}\left[\tanh\left(9y-9x\right)+1\right]$$
$$f_3(x,y) = \frac{1.25+\cos\left(5.4\,y\right)}{6(1+(3\,x\,-\,1)^2)}$$
$$f_5(x,y) = \frac{\exp\left(-81/16\left[(x-0.5)^2+(y-0.5)^2\right]\right)}{3}$$
$$f_5(x,y) = \frac{\exp\left(-81/4\left[(x-0.5)^2+(y-0.5)^2\right]\right)}{3}$$
$$f_6(x,y) = \frac{1}{9}\sqrt{(64-81\left[(x-0.5)^2+(y-0.5)^2\right)]} - 0.5$$

References

[1] V. Bayona, M. Moscoso, M. Carretero, M. Kindelan, RBF-FD formulas and convergence properties, J. Comput. Phys. 229 (2010) 8281-8295.

[2] P. González-Rodríguez, V- Bayona, M. Moscoso, and M. Kindelan, Laurent series based RBF-FD method to avoid ill-conditioning, Engineering Analysis with Boundary Elements 52 (2015) 24-31.

[3] P. González-Rodríguez, M. Moscoso, and M. Kindelan, Laurent expansion of the inverse of perturbed, singular matrices, Journal of Computational Physics 299 (2015) 307-319.

[4] M. D. Buhmann, S. Dinew, Limits of radial basis function interpolants, Communications on Pure and Applied Analysis, 6 (2007) 569-585.

[5] R. E. Carlson, T. A. Foley, The parameter R^2 in multiquadric interpolation, Comput. Math. Appl. 21 (1991) 29-42.

[6] T. A. Driscoll, B. Fornberg, Interpolation in the limit of increasingly flat radial basis functions, Comput. Math. Appl. 43 (2002) 413-422.

[7] G. E. Fasshauer, Meshfree Approximation Methods with MATLAB, World Scientific Publishing Co., Singapore (2007).

[8] G. E. Fasshauer, J. G. Zhang, On choosing optimal shape parameters for RBF approximation, Numer. Algor., 45 (2007) 345-368.

[9] G. E. Fasshauer, M. J. McCourt, Stable evaluation of Gaussian RBF interpolants, SIAM J. Sci. Comput., 34 (2012), 73-762.

[10] N. Flyer, E. Lehto, Rotational transport on a sphere: Local node refinement with radial basis functions, J. Comput. Phys. 229 (2010) 1954-1969.

[11] N. Flyer, B. Fornberg, Radial basis functions: developments and applications to planetary scale flows, Comput. & Fluids, 46 (2011), 23-32.

[12] B. Fornberg, G. Wright, Stable computation of multiquadric interpolation for all values of the shape parameters, Comput. Math. Appl. 48 (2004) 853-867.

[13] B. Fornberg, G. Wright, E. Larsson, Some observations regarding interpolants in the limit of flat radial basis functions, Comput. Math. Appl., 47 (2004) 37-55.

[14] B. Fornberg, C. Piret, A stable algorithm for flat radial basis functions on a sphere. SIAM J. Sci. Comput., 30 (2007) 60-80.

[15] B. Fornberg, E. Larsson, N. Flyer, Stable computations with Gaussian radial basis functions, SIAM J. Sci. Comput. 33 (2011) 869-892.

[16] B. Fornberg, E. Lehto, C. Powell, Stable calculation of Gaussian-based RBF-FD stencils, Comput. Math. Appl. 65 (2013) 627-637.

[17] R. Franke, A critical comparison of some methods for interpolation of scattered data, Naval Postgraduate School, Technical Report NPS-53-79-003 (1979).

[18] E. Larsson, B. Fornberg, Theoretical and computational aspects of multivariate interpolation with increasingly flat radial basis functions, Comput. Math. Appl. 49 (2005) 103-130.

[19] Y. J. Lee, G. J. Yoon, J. Yoon, Convergence of Increasingly Flat Radial Basis Interpolants to Polynomial Interpolants, SIAM J. Math. Anal. 39 (2007) 537-553.

[20] Y. J. Lee, G. J. Micchelli, J. Yoon, On Convergence of Flat Multivariate Interpolation by Translation Kernels with Finite Smoothness, Constr. Approx. 40 (2014) 37-60.

[21] Y. J. Lee, C. A. Micchelli, J. Yoon, A study on multivariate interpolation by increasingly flat kernel functions, J. Math. Anal. Appl 427 (2015) 74-87.

[22] Randall J. LeVeque, Finite Difference Methods for Ordinary and Partial Differential Equations: Steady-State and Time-Dependent Problems, SIAM (2007).

[23] C. A. Micchelli, Interpolation of scattered data: Distance matrices and conditionally positive definite functions, Constr. Approx. 2, (1986) 11-22.

[24] S. Rippa, An algorithm for selecting a good value for the parameter c in radial basis function interpolation, Adv. Comput. Math. 11 (1999) 193-210.

[25] R. M. C. Roque, A. J. M. Ferreira, Numerical experiments on optimal shape parameters for radial basis functions Numer. Methods Partial Differ. Equ., 26 (2010) 675-689.

[26] Y. V. S. S. Sanyasiraju, C. Satyanarayana On optimization of the RBF shape parameter in a grid-free local scheme for convection dominated problems over non-uniform centers, Appl. Math. Model., 37 (2013) 7245-7272

[27] R. Schaback, Multivariate interpolation by polynomials and radial basis functions, Constr. Approx., 21 (2005) 293-317.

[28] R. Schaback, Limit problems for interpolation by analytic radial basis functions, J. Comp. Appl. Math. 212 (2008) 127-149.

[29] M. Scheuerer, An alternative procedure for selecting a good value for the parameter c in RBF-interpolation, Adv. Comput. Math. 34 (2011) 105-126.

[30] M. Uddin, On the selection of a good value of shape parameter in solving time-dependent partial differential equations using RBF approximation method, Appl. Math. Model.

[31] G. B. Wright, B. Fornberg, Scattered node compact finite difference-type formulas generated from radial basis functions, J. Comput. Phys. 212 (2006) 99-123.

[32] Z. Wu and R. Schaback, Local Error Estimates for Radial Basis Function Interpolation of Scattered Data, IMA J. Numer. Anal., 13 (1993), 13-27.